# 机器学习导论
# 习题五

151250104, 卢以宁, kiwiloveskiwis@gmail.com

2017 年 5 月 31 日

## 1 [25pts] Bayes Optimal Classifier

试证明在二分类问题中，但两类数据同先验、满足高斯分布且协方差相等时，LDA 可产生贝叶斯最优分类器。

**Solution.** 由题意, $p(c=0)=p(c=1), p(\mathbf{x}|c=0)=N(\boldsymbol{\mu_1},\Sigma), p(\mathbf{x}|c=1)=N(\boldsymbol{\mu_2},\Sigma)$
则贝叶斯最佳判别面为:

$$\begin{aligned} p(c=0|\mathbf{x}) &= p(c=1|\mathbf{x}) \\ \ln[p(c=0)p(\mathbf{x}|c=0)] &= \ln[p(c=1)p(\mathbf{x}|c=1)] \\ (\mathbf{x}-\boldsymbol{\mu_1})^{\mathbf{T}}\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu_1}) &= (\mathbf{x}-\boldsymbol{\mu_2})^{T}\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu_2}) \\ \Sigma^{-1}(\boldsymbol{\mu_1}-\boldsymbol{\mu_2})\mathbf{x} &= \frac{1}{2}(\mu_1^T\Sigma^{-1}\mu_1-\mu_2^T\Sigma^{-1}\mu_2) \end{aligned} \tag{1.1}$$

令 $\mathbf{w}$ 为 $\Sigma^{-1}(\boldsymbol{\mu_1}-\boldsymbol{\mu_2})$, $b$ 为 $-\frac{1}{2}(\mu_1^T\Sigma^{-1}\mu_1-\mu_2^T\Sigma^{-1}\mu_2)$ 即可。接下来证明使用 LDA 将得到相同结果。

$$\begin{aligned} \min_{\mathbf{w}} \quad & -\mathbf{w^T S_b w} \\ \text{s.t.} \quad & -\mathbf{w^T S_w w}=1 \end{aligned} \tag{1.2}$$

解得

$$\begin{aligned} \mathbf{w} &= \mathbf{S_w^{-1}}(\boldsymbol{\mu_0}-\boldsymbol{\mu_1}) \\ &= \sum_{x\in X_0}(\mathbf{x}-\boldsymbol{\mu_0})(\mathbf{x}-\boldsymbol{\mu_0})^{\mathbf{T}} + \sum_{x\in X_1}(\mathbf{x}-\boldsymbol{\mu_1})(\mathbf{x}-\boldsymbol{\mu_1})^{\mathbf{T}} \\ &= 2\Sigma^{-1}(\boldsymbol{\mu_1}-\boldsymbol{\mu_2}) \end{aligned} \tag{1.3}$$

则投影中心分别为 $\mathbf{w}\boldsymbol{\mu_0}$ 和 $\mathbf{w}\boldsymbol{\mu_1}$. 故 $b=-\frac{1}{2}(\mathbf{w}\boldsymbol{\mu_0}+\mathbf{w}\boldsymbol{\mu_1})$. 即 LDA 可以得到贝叶斯最优判别面。

## 2 [25pts] Naive Bayes

考虑下面的 400 个训练数据的数据统计情况，其中特征维度为 2 ( $\mathbf{x}=[x_1,x_2]$ )，每种特征取值 0 或 1，类别标记 $y\in\{-1,+1\}$。详细信息如表1所示。

根据该数据统计情况，请分别利用直接查表的方式和朴素贝叶斯分类器给出 $\mathbf{x} = [1, 0]$ 的测试样本的类别预测，并写出具体的推导过程。

表 1: 数据统计信息

| $x_1$ | $x_2$ | $y = +1$ | $y = -1$ |
|---|---|---|---|
| 0 | 0 | 90 | 10 |
| 0 | 1 | 90 | 10 |
| 1 | 0 | 51 | 49 |
| 1 | 1 | 40 | 60 |

**Solution.** (1) 查表方式:

$$P(y = +1|x = [1, 0]) = 0.51 \tag{2.1}$$

故预测 $y = 1$.

(2) 朴素贝叶斯:

$$P(y = 1) = \frac{271}{400}, P(y = -1) = \frac{129}{400} \tag{2.2}$$

$$\begin{aligned}
P(x_1 = 0|y = 1) = \frac{180}{271}, P(x_1 = 0|y = -1) = \frac{20}{129} \\
P(x_1 = 1|y = 1) = \frac{91}{271}, P(x_1 = 1|y = -1) = \frac{109}{129} \\
P(x_2 = 0|y = 1) = \frac{141}{271}, P(x_2 = 0|y = -1) = \frac{59}{129} \\
P(x_2 = 1|y = 1) = \frac{130}{271}, P(x_2 = 1|y = -1) = \frac{70}{129}
\end{aligned} \tag{2.3}$$

$$\begin{aligned}
\frac{P(y = +1|x_1 = 1, x_2 = 0)}{P(y = -1|x_1 = 1, x_2 = 0)} &= \frac{P(y = +1) \times P(x_1 = 1|y = +1) \times P(x_2 = 0|y = +1)}{P(y = -1) \times P(x_1 = 1|y = -1) \times P(x_2 = 0|y = -1)} \\
&= \frac{271 \cdot 91 \cdot 141}{400 \cdot 271 \cdot 271} \Big/ \frac{129 \cdot 109 \cdot 59}{400 \cdot 129 \cdot 129} = \frac{165199}{1742801} < 1
\end{aligned} \tag{2.4}$$

故预测 $y = -1$.

## 3 [25pts] Bayesian Network

贝叶斯网 (Bayesian Network) 是一种经典的概率图模型，请学习书本 7.5 节内容回答下面的问题：

(1) **[5pts]** 请画出下面的联合概率分布的分解式对应的贝叶斯网结构：

$$\Pr(A, B, C, D, E, F) = \Pr(A)\Pr(B)\Pr(C)\Pr(D|A)\Pr(E|A)\Pr(F|B, D)\Pr(G|D, E)$$

(2) **[5pts]** 请写出图3中贝叶斯网结构的联合概率分布的分解表达式。

(3) **[15pts]** 基于第 (2) 问中的图3, 请判断表格2中的论断是否正确，只需将下面的表格填完整即可。

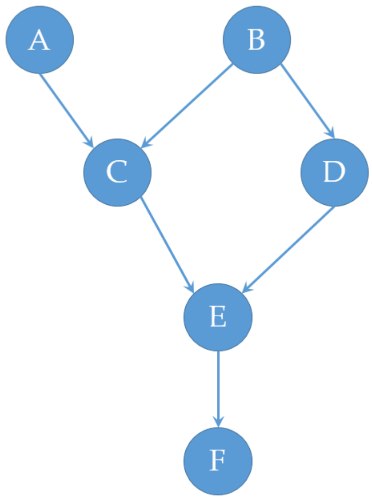
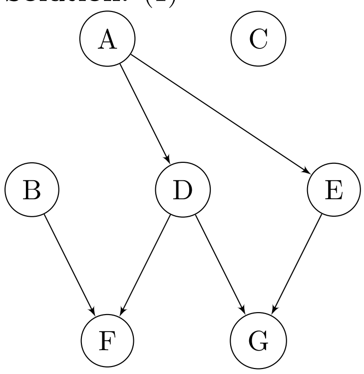



图 1: 题目 3-(2) 有向图

表 2: 判断表格中的论断是否正确

| 序号 | 关系 | True/False | 序号 | 关系 | True/False |
|---|---|---|---|---|---|
| 1 | $A \perp\!\!\!\perp B$ | True | 7 | $F \perp B \vert C$ | False |
| 2 | $A \perp B \vert C$ | False | 8 | $F \perp B \vert C, D$ | True |
| 3 | $C \perp\!\!\!\perp D$ | False | 9 | $F \perp B \vert E$ | True |
| 4 | $C \perp D \vert E$ | False | 10 | $A \perp\!\!\!\perp F$ | False |
| 5 | $C \perp D \vert B, F$ | False | 11 | $A \perp F \vert C$ | False |
| 6 | $F \perp\!\!\!\perp B$ | False | 12 | $A \perp F \vert D$ | False |

**Solution.** (1)




(2)

$$\Pr(A, B, C, D, E, F) = \Pr(A)\Pr(B)\Pr(C|A, B)\Pr(D|B)\Pr(E|C, D)\Pr(F|E)$$

## 4 [25pts] Naive Bayes in Practice

请实现朴素贝叶斯分类器，同时支持离散属性和连续属性。详细编程题指南请参见链接：http://lamda.nju.edu.cn/ml2017/PS5/ML5_programming.html.

同时，请简要谈谈你的感想。实践过程中遇到了什么问题，你是如何解决的？

**Solution.** 主要问题在于对于数据集 v0.0 不考虑连续变量时精度上升了将近一倍, 并且快了许多许多。我很想因此把连续变量忽略掉, 但忍住了; 结论是 tf-idf 不适合用 Naive Bayes. 但是更改后的数据集一劳永逸地解除了烦恼, (即使研究发现连续属性占了主导地位.)
至于连续属性有多举足轻重, 我们进行了一个实验: 每次预测时, 从后 2500 维数据算出 continuous_prob_sum 从前 2500 维数据算出 discrete_prob_sum ，将它们的比例存下来.

```
RATIO = 1 # we don't know it yet


for ins in range(0, data.shape[0]):
        for cval in range(num_of_class): # class_value
                discrete_prob_sum = 0
                continuous_prob_sum = 0
                for i in range(0, 5000):
                        if(i < 2500):
                                discrete_prob_sum += ...
                        else:
                                continuous_prob_sum += ...
                ratio[ins, cval] = continuous_prob_sum / discrete_prob_sum
                class_prob[cval] = np.log(cmat.shape[0] / data.shape[0])
                          + discrete_prob_sum + continuous_prob_sum * RATIO
        predict_y[ins] = np.argmax(class_prob)
.....
```

得到 ratio 的结果如下 (针对测试集):

```
>>> np.mean(ratio, axis=0)
array([ 2706665.65332563,  4083342.44150028,  4219538.94068223,
        3734542.59047444,  2785861.71508093])
>>> np.mean(ratio)
3505990.2682127017
```

对于训练集:

```
>>> np.mean(ratio, axis=0)
array([ 2703279.99020332,  3436356.2750135 ,  3283160.81424551,
        3163941.63246489,  1676037.38363671])
>>> np.mean(ratio)
2852555.2191127865
```

简单起见，我们使用了训练集全局的平均值 2852555, 从而 `RATIO = 1 / 2852555`. 在计算总 prob 时, 进行加权求和。
注意到当我们仅使用离散属性进行预测时, 准确率为 0.6639; 仅使用连续属性/两者皆使用进行预测时, 准确率为 0.6844. 而在按上述方法预测时, 准确率达到 0.7152.
真是要歌功颂德了呀。